Multimedia Short Text Classification via Deep RNN-CNN Cascade

taoaishan/ per 1st Affiliation (Author)
1531844@tongji.edu.cn

huchaochao/ per 2nd Affiliation (Author)
1631580@tongji.edu.cn

***Abstract*—With the rapid development of mobile technologies, social networking softwares such as Twitter, Weibo and WeChat are becoming ubiquitous in our every day life. These social networks generate a deluge of data that consists of not only plain texts but also images, videos, and audios. As a consequence, the traditional approaches that classify the short text by counting only the key words become inadequate. In this paper, we propose a multimedia short text classification approach by deep RNN(Recurrent neural network) and CNN(Convolutional neural network) cascade. We first employ an LSTM(Long short-term memory) network to convert the information in the images into text information. Then a convolutional neural network is used to classify the multimedia texts by taking into account both the texts generated from the image as well as those contained in the initial message. It is seen through experiments using MSCOCO dataset that the proposed method exhibits significant performance improvement over the traditional methods.**

I. Introduction

In the era of big data, the traditional form of information has been changed. Most obviously, the long text form has already been replaced by the short text form. For example, we can use Twitter to publish our messages with limited 140 words. Meanwhile, the plain texts has been gradually replaced by messages of images and short plain texts. There is an instance from WeChat's Circles:“ Today is a special day for every American!” with a picture that draws Trump and Hillary. If we just use picture and text separately, we can get no conclusion. But if content was extracted from this image and combine the content with original text, we will know this message should belong to the category of politics. And we may draw a conclusion that today is the presidential election day.

Traditionally, there were two common methods to classify multimedia short texts. The first method is to classify the raw data by image information without considering the text parts [1]. The second method is to classify the raw data by text information without considering the image parts. There are two types of text classifiers. One is text classifier such as mixture modeling text classifier, probabilistic and naive Bayes classifier [2], [3] and so on. The other is CNN(Convolutional neural network) text classifier which is based on deep learning [4], [5].

In this paper, we try to combine original texts and image contents to classify multimedia short texts. How to extract useful information from image? What rules should obey when we extract information from images? We propose a new RNN-CNN model to answer these two problems and obtains a better result.

The remainder of our paper is organized as follows: we review some related works in section 2. We give the architecture of proposed model and clearly describe the details of the model in section 3. The section 4 demonstrates effective result of the proposed model with experiments. At last, we make a whole summary of this paper.

II. Related Works

With the advent of different social softwares, the messages are organized in a variety of ways and organised by many new features, such as “@username” feature [6] ,slangs, percentage signs and so on.It is difficult for traditional models to summarize so many data features.The traditional input form— bag of words(BOW) comes across data sparsity problem [6], [7], [8]. The text classifiers which are based on deep learning are getting more and more popular. Word representation vectors as the input of neural network exert great influence on classifying results. The effective pre-trained word embeddings can be obtained according to neural network [9], [10], [11]. Not only word embedding vectors keep the relationships of words, but also satisfy the context semantic relationships.

As described before, images may have a strong association with original texts in most instances when they are in the same multimedia text. Recently, some images classification methods which depend on deep learning have got an improved achievement [12], [13], so that we can get its category just depending on images content. For example, when there is a picture painted with a solider, there is a high probability that it belongs to military category. In this paper we take images information into account when classifying multimedia short texts by methods based on deep learning. If we want to combine original texts and image contents together as a new input of CNN text classifiers, they must be projected to the same vector space so that they can be used as the unified input of CNN classifiers. In addition, when we want to extract useful sentences from pictures, these sentences should be grammatical and consistent with the picture content. There are lots of models about image caption work [14], [15]. But they all encounter data sparsity problem in the big data field. Recently, some special RNN (Recurrent neural network) models are applied to the machine translation field and make important achievements [16], [17], [18].

With the development of computer vision [19], the accuracy of image classification has arrived at a high level. These image recognition methods can not describe the content of pictures effectively when images are blurry.

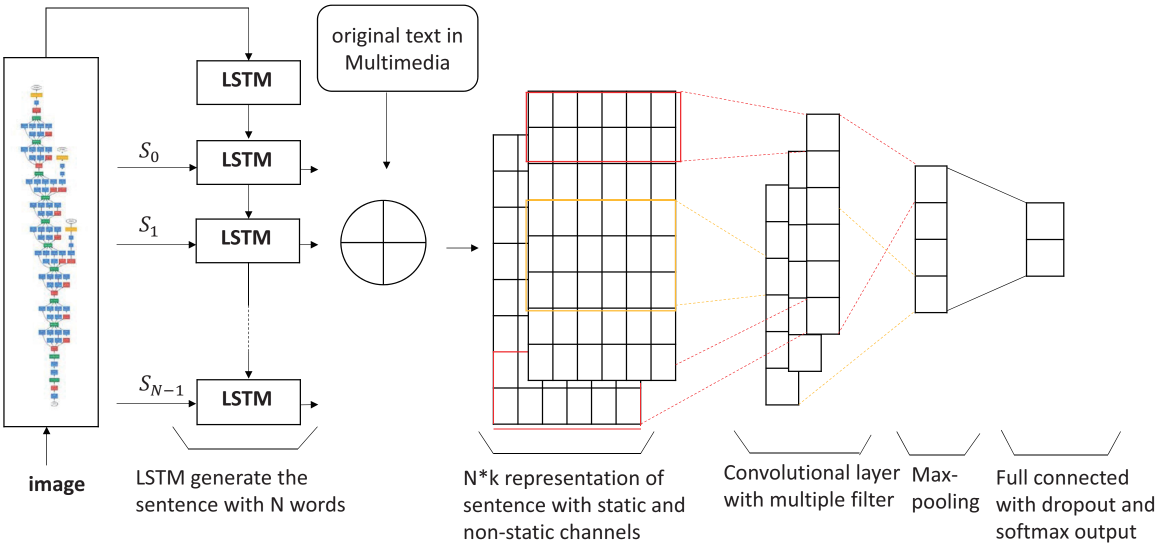
original text in Multimedia
LSTM
LSTM
LSTM
LSTM
S_0
S_1
S_{N-1}
image
LSTM generate the sentence with N words
N*k representation of sentence with static and non-static channels
Convolutional layer with multiple filter
Max-pooling
Full connected with dropout and softmax output

Figure 1. This is an inserted RNN-CNN graphic

In this paper we apply LSTM model [20] to generate a sentence for describing an image. Even if the images are blurry, the sentences can also give some relative verbs and adjectives which are helpful for classifying images. The core of LSTM model is to predict every word in sentences correctly. The Kiros team [21] used a convolutional neural network to predict next word based on given sentences and pictures. There are some other works which applied recurrent neural network to the prediction tasks [22], [23].

III. RNN-CNN Cascade Model

A. Overall of RNN-CNN

Our model consists of an image caption model and a CNN text classifier. At first, we use a pre-trained CNN model to encode an image I as a fix-length vector [24]. The "show and tell" model [20] will be adopted to generate sentences subsequently. The objective function $p(\boldsymbol{S} \mid \boldsymbol{I})$ is taken to train the LSTM model, where S represents the generated sentence. The correct sentence representation of an image will be obtained when $p(\boldsymbol{S} \mid \boldsymbol{I})$ arrives at maximum value. Then we connect this sentence with an original short text as the CNN text classifier input. This end-to-end model is called RNN-CNN(Recurrent neural network cascade with Convolutional neural network) classifiers as in Figure 1.

In the process of generating descriptions, the model have to meet two requirements which have been described in Introduction. The RNN model is selected to generate descriptions because the RNN model can save contextual relationships.

For a given image in the multimedia short text, the best image description can be computed by conditional probability as follows.

$$p(\boldsymbol{S} \mid \boldsymbol{I}) = \prod_{t=1}^{T} p(\boldsymbol{s}_t|\boldsymbol{I}, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{t-1}) \tag{1}$$

Where $\boldsymbol{S}$ = $(\boldsymbol{s}_1, \cdots, \boldsymbol{s}_T)$ represents the sentence generated by RNN model. $\boldsymbol{s}_t$,$t = 1, \cdots, T$,represents a word in the sentence $\boldsymbol{S}$. T is the number of the words in the sentence and it is an unbound value. $\boldsymbol{I}$ is an input image. In the process of RNN , the model generates a single word $\boldsymbol{s}_t$ at each iteration. We employ GoogLeNet pre-trained model on ILSVRC 2014 dataset for object recognition and detection [25], which is the largest image classification dataset at present. The iteration will be continued unless it generates the character of <EOS>.

B. LSTM Model

Each iteration of RNN has three inputs and one output, t represents the numbers of iterations. One of the inputs is the memory $\boldsymbol{h}_t$, which is a hidden state with fixed length. The memory $\boldsymbol{h}_t$ saves information that generated from beginning to end and updated at each time step with another input $\boldsymbol{x}_t$. The last input is the memory cell state $\boldsymbol{c}_{t-1}$. The output of each iteration is probability distribution over all words.

$$\boldsymbol{h}_{t+1} = f(\boldsymbol{h}_t, \boldsymbol{x}_t) \tag{2}$$

We choose LSTM as $f(\cdot)$. The LSTM is designed to avoid long-term dependency problem and it has carefully been designed for the structure of gates. The gate is a method which controlling information from $\boldsymbol{h}_{t-1}$. The formulae to calculate the memory cell output and update the memory cell state through gates are as follows:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{W}_{fh}\boldsymbol{h}_t) \tag{3}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_t) \tag{4}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot tanh(\boldsymbol{W}_{cx}\boldsymbol{x}_t + \boldsymbol{W}_{ch}\boldsymbol{h}_t) \tag{5}$$

$$\boldsymbol{o}_t = \sigma(\ \boldsymbol{W}_{ox}\boldsymbol{x}_t + \boldsymbol{W}_{oh}\boldsymbol{h}_t) \tag{6}$$

$$\boldsymbol{h}_{t+1} = \boldsymbol{o}_t \odot tanh(\boldsymbol{c}_t) \tag{7}$$

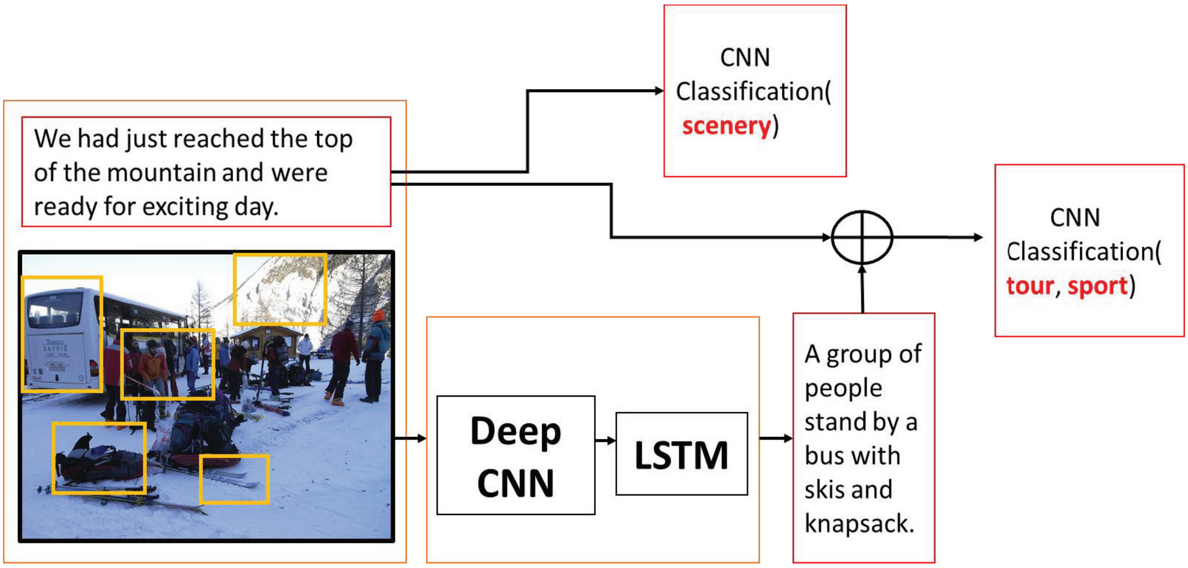
We had just reached the top of the mountain and were ready for exciting day.
CNN Classification(scenery)
CNN Classification(tour, sport)
Deep CNN
LSTM
A group of people stand by a bus with skis and knapsack.

Figure 2. This is an example of multimedia short text application

$$\boldsymbol{p}_{t+1} = Softmax(\boldsymbol{h}_{t+1}) \tag{8}$$

σ is a sigmoid function. $\odot$ is an element-wise multiplication operation.The various $\boldsymbol{W}$ matrices are trained parameters.The forget gate $\boldsymbol{f}_t$ decides how much information should be dropped. The value of $\boldsymbol{f}_t$ is between 0 (drop completely)and 1 (keep completely). The input gate $\boldsymbol{i}_t$ controls updating process. It decides which part of the input $\boldsymbol{x}_t$ and the last hidden state $\boldsymbol{h}_t$ should be updated to the memory cell $\boldsymbol{c}_t$. The output gate $\boldsymbol{o}_t$ decides which part of a memory cell $\boldsymbol{c}_t$ should be outputted. The $tanh$ is a hyperbolic tangent function. The $\boldsymbol{p}_{t+1}$ represents the probability distribution over all words. The initialized input of LSTM is an image which is disposed by CNN. The image $\boldsymbol{I}$ is only input once.

$$\boldsymbol{x}_{-1} = CNN(\boldsymbol{I}) \tag{9}$$

The output sequence looks like $(\boldsymbol{s}_1 \cdots \boldsymbol{s}_N)$. Our loss is the sum of the negative log likelihood of the correct word at each step as follows:

$$L = -\sum_{t=1}^{N} \log \boldsymbol{p}_t(\boldsymbol{s}_t) \tag{10}$$

$\boldsymbol{p}_t$ represents the probability of generating correct word $\boldsymbol{s}_t$. When sum of these probabilities arrives at the largest level, the loss function arrives at the minimal value. When getting the sentences generated by LSTM model, we have to use some metrics to evaluate the generated sentences [26]. Now we use the CIDEr [27] as main metric and use the BLEU [28] as auxiliary metric. The higher score the sentence get, the more effective they are, both based on CIDEr and BLEU.

C. CNN Model

The following is the whole process of RNN-CNN model (Figure 1).The first layer is an input layer. The sentence of $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_N\}$ is generated by LSTM model. N represents word numbers in the longest sentence. $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_{T'}\}$ is the original caption in a multimedia short text. Now we connect them as $\boldsymbol{P} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_n\}$,$\boldsymbol{P} = \boldsymbol{S} \bigoplus \boldsymbol{Y}$. The n equals the sum of N plus T'. We apply word embedding from scratch to represent each word in the sentence. For the sentence $\boldsymbol{P} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_n\}$, it is projected to a matrix $\boldsymbol{M}$. $\boldsymbol{M} \in \boldsymbol{R}^{n*k}$ is obtained by looking up table. k is the dimension of word embedding. Every word in the sentence be the k-dimensional word vector.

In the convolution layer, we use the $\boldsymbol{r}_{i:i+j-1}$ to represent concatenation of words $\boldsymbol{r}_i, \boldsymbol{r}_{i+1}, \cdots, \boldsymbol{r}_{i+j-1}$.$j$ is the window kernel size of convolution. The weight $\boldsymbol{W}^{'} \in \mathbb{R}^{n*k}$ is applied to the convolution operation to produce a new feature. The process of generating new feature c_i is as follows:

$$c_i = f(\boldsymbol{W}^{'} \cdot \boldsymbol{r}_{i:i+j-1} + b) \tag{11}$$

f is a non-linear function . $b \in \mathbb{R}$ is a bias term. We apply the kernel with window size j to slide sentence $\boldsymbol{P} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_n\}$ for getting a feature map:

$$\boldsymbol{c} = [c_1, c_2, \cdots, c_{n-j+1}] \tag{12}$$

$\boldsymbol{c} \in \mathbb{R}^{n-j+1}$ is generated with a window kernel size of j.The main purpose of convolutional layer is to strengthen the local features. The different convolutional kernels with different sizes can extract multiple features.

The following is max pooling layer. The max pooling operation select the maximum value $\hat{c} = max\{\mathbf{c}\}$ on $\boldsymbol{c}$ as the feature. The maximum value can capture the most obvious feature of each feature map.The max pooling layer can reduce parameters number and decrease time complexity of whole model. It also can solve the problems of variable sentence lengths .

The next layer is fully connected softmax layer with l_2-norms dropout operation. Some neural units are ignored for preventing co-adaptation of hidden units. After being disposed by the max pooling layer, the output is $\boldsymbol{z} = [\hat{c}_1, \cdots, \hat{c}_m]$. m represents the number of filters. The process of dropout is as following:

$$\boldsymbol{h} = \boldsymbol{W}^{''} \cdot (\boldsymbol{q} \odot \boldsymbol{z}) + b \tag{13}$$

$\boldsymbol{h}$ is a output unit in forward propagation.$\boldsymbol{W}^{''}$ is weight matrix. $\odot$ is the element-wise multiplication operator. $\boldsymbol{q} \in \mathbb{R}^m$ is a 'masking' vector which all elements' values in it are coincidence of Bernoulli variables. This dropout operation mainly for avoiding overfitting. Meanwhile, the Softmax operation is applied as a classifier to generate the probability distribution over categories.

Our experiment takes multichannel architecture, which showed in Figure 1. One is static channel, which these parameters keep static in process of training. The other is non-static channel, which these parameters can be fine-tuned via backpropagation. The multichannel has better performance on many datasets. Because the fine-tune operation makes vectors more close to specific duty. At the same time, the non-static operation limits the changing distance of vectors.Figure 2 shows an example of multimedia short text application. As we can see, our model can do a better classification than the traditional CNN model.

IV. Experiment

MSCOCO [29] is a suitable dataset for multimedia short text classifying. Because MSCOCO contains 91 categories and each category consists of many multimedia short texts. For testing the robustness of our model, we conduct three groups of experiments to test the result respectively based on two-category, three-category and five-category dataset(Figure 3).

The two-category experiment is based on vehicle category which includes five subclasses (car, boat, truck, airplane, train) and animal category which includes five subclasses (giraffe, zebra, bear, sheep, elephant). The three-category dataset is based on the two-category dataset and furniture category dataset which includes five subclasses (bed, couch, closestool, table, chair). The five-category dataset added sports category dataset which includes five subclasses (skis,snowboard,kite,baseball,skateboard) and food category dataset which includes five subclasses (sandwich,cake,orange,broccoli,carrot) on the basis of three-category dataset. In order to ensure the fairness and accuracy of our experiment, we extract the 1000 multimedia short texts of each class for training, each subclass has 200 multimedia short texts. At the same time, we extract the 300 multimedia short texts of each class for testing, each subclass has 60 multimedia short texts.

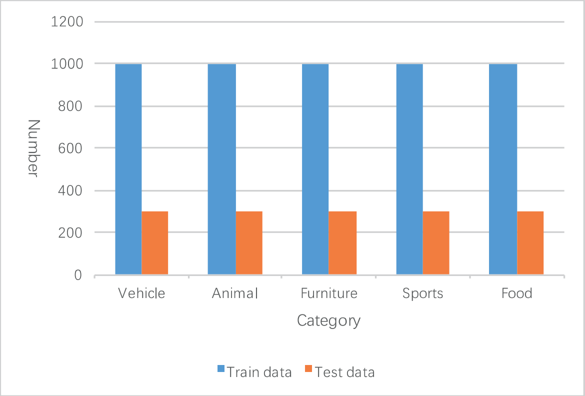

Figure 3. This is the dataset graph for training and testing

set method	Two-Category	Three-Category	Five-Category
text(CNN)	0.86512	0.80000	0.76594
image(ResNet34)	0.90625	0.84961	0.80097
image-text(RNN-CNN)	0.95116	0.95504	0.94778

Table I
THE CLASSIFICATION ACCURACY OF PROPOSED RNN-CNN METHOD AGAINST OTHER MODELS

We use the caption texts and their labels together to train and test with text CNN. Then we apply images and their labels together to train and test with image ResNet. At last, the RNN-CNN model (the CIDEr value is 85) does a new classification experiment by means of using multimedia short texts. Then we compare these three cases' results carefully (Table 1).

As we can see, the classification accuracy of image ResNet34 model is four percent higher than the average of text CNN model. But the classification accuracy of image-text RNN-CNN model is more higher than the image ResNet34 model. Meanwhile, the accuracy of image ResNet34 and text CNN model decreases as classification quantity increases, while image-text RNN-CNN model maintains almost the same accuracy on the two-category and multi-category classification. The RNN-CNN model has higher stability and robustness.

V. Conclusion

This paper takes new data pattern into consideration and proposes an end to end novel classifier model that can automatically classify the new media data. The traditional image classifier can capture image features. But the image can not be recognized correctly when it's blurry. When changing image to sentence, we can get helpful semantic relationship like some verbs and adjectives which are very useful for classifying. When the generated sentence connects with the original description together as input, as our experiment shows, the accuracy and stability of RNN-CNN model all get great improvement. The applicability of RNN-CNN model will be extensive so that it can be widely applied to data classification of new media.

Acknowledgment

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] C. C. Aggarwal and C. Zhai, “A survey of text classification algorithms,” in *Mining text data*. Springer, 2012, pp. 163–222.

[3] C. C. Aggarwal, S. C. Gates, and P. S. Yu, “On using partial supervision for text categorization,” *IEEE Transactions on Knowledge and data Engineering*, vol. 16, no. 2, pp. 245–255, 2004.

[4] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[5] P. Wang, J. Xu, B. Xu, C.-L. Liu, H. Zhang, F. Wang, and H. Hao, “Semantic clustering and convolutional neural network for short text categorization.” in *ACL (2)*, 2015, pp. 352–357.

[6] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, “Short text classification in twitter to improve information filtering,” in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 841–842.

[7] M. Chen, X. Jin, and D. Shen, “Short text classification improved by learning multi-granularity topics,” in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[8] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, “Learning to classify short and sparse text & web with hidden topics from large-scale data collections,” in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 91–100.

[9] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[14] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, “Every picture tells a story: Generating sentences from images,” *Computer vision–ECCV 2010*, pp. 15–29, 2010.

[15] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, “Babytalk: Understanding and generating simple image descriptions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.

[16] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[20] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[21] R. Kiros, R. Salakhutdinov, and R. Zemel, “Multimodal neural language models,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 595–603.

[22] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, “Explain images with multimodal recurrent neural networks,” *arXiv preprint arXiv:1410.1090*, 2014.

[23] R. Kiros, R. Salakhutdinov, and R. S. Zemel, “Unifying visual-semantic embeddings with multimodal neural language models,” *arXiv preprint arXiv:1411.2539*, 2014.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229*, 2013.

[25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[26] M. Hodosh, P. Young, and J. Hockenmaier, “Framing image description as a ranking task: Data, models and evaluation metrics,” *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.

[27] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “Cider: Consensus-based image description evaluation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4566–4575.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.